behavior contract of move with god card

move(x: int, y: int): Game

Preconditions:

1. Game state conditions:
   - currentPhase = MOVE
   - winner = null (game not ended)
   - selectedWorker ≠ null

2. Ownership conditions: the owner of selectedWorker must be current player
   - selectedWorker.owner = players[currentPlayerIndex]

3. Move validity conditions:
   - Space at (x, y) $\in$ getAvailableMoves(board, selectedWorker)
     That is:
     1. destSpace has no dome
     2. destSpace tower level - currentSpace tower level <=1
     3. destSpace is not occupied by worker or satisfy special god card condition

4. God card specific preconditions for move:
   - if godCards[currentPlayerIndex] == Minotaur:
     Space(x,y) occupied by a worker is available for move if:
     1. Space(x,y).occupant.owner ≠ selectedWorker.owner (opponent worker)
     2. Push space exists and is valid:
        - pushRow = x + (x - selectedWorker.position.row)
        - pushCol = y + (y - selectedWorker.position.col)
        - 0 ≤ pushRow < 5 AND 0 ≤ pushCol < 5
        - Space(pushRow, pushCol).occupant = null
        - Space(pushRow, pushCol).hasDome = false
        - Space(pushRow, pushCol) tower level - Space(x,y) tower level <=1

Postconditions (must be true AFTER move)):

1. Worker position updated:
   - selectedWorker.position = Space(x, y)
   - Space(x, y).occupant = selectedWorker
   - old Space(selectedWorker.position).occupant = null

2. God card specific postconditions:
   - if Minotaur push occurred:
     • opponentWorker.position = Space(pushRow, pushCol)
     • Space(pushRow, pushCol).occupant = opponentWorker
     • Space(x, y).occupant = selectedWorker (after push)

3. Board state updated:
   - board' = board.updateSpace(oldPosition).updateSpace(newPosition)
   - IF Minotaur: board' includes updateSpace(pushPosition)
   - All other spaces unchanged

4. Phase transition (normal case):
   - IF not met winCondition and loseCondition:
     • currentPhase' = BUILD
     • buildableSpaces' = getAvailableBuilds(board', selectedWorker, null)
     • selectedWorker' = selectedWorker (preserved)
     • currentPlayerIndex' = currentPlayerIndex (unchanged)

5. Win condition handling:
   - IF selectedWorker moved to level 3:
     • winner' = currentPlayerIndex
     • currentPhase' = END_GAME

6. Loss condition handling:

   - IF buildableSpaces' = ∅:

     • winner' = 1 - currentPlayerIndex (opponent wins)
     • currentPhase' = END_GAME